TIMEX
INTER-
OFFICE
MEMO

TO: Distribution

DATE: October 12, 1983

FROM: S. McCourt

SUBJECT: TIMEX MICRODRIVE OPERATING SYSTEM

DISTRIBUTION: Boyle, Branigin, Collum, Corcoran, Dargan, Galie,
Grimm, Hildreth, Madden, Shiff, Smith

Attached is the Timex Microdrive Operating System for your review. It reflects the shift in priorities from floppy disk to microdrive based product (including support for the RS-232 channel and local network). Please direct comments or questions to me at Ext. 6313.

SMc:js
attachment

Timex Microdrive Operating System
Functional Specification

Scott McCourt

10/10/83

Computer Engineering

Timex Computer Corporation

Middlebury, CT

Table of Contents

1.0 INTRODUCTION

This document describes the functionality of the system software which controls the following devices on the TS 2000:

a) one or more microdrives

b) an RS 232 interface

c) a local area network

The commands in Section 5.0 are to be implemented as a set of commands in BASIC.

When the operating system first boots up, the user is prompted for the current date. The format to be used when entering the date is mm-dd-yy. An invalid date will produce an error message and the user will be prompted to re-enter the date. This date is not updated, but is used only for file directory information (date created, date last updated). If nothing is entered (i.e., the null string) the date is set to be 00-00-00.

2.0 RELATED DOCUMENTS

[1] TS 2000 System Software Functional Specification

[2] TS 2000 System Software Architecture Specification

3.0 FILE NAMING CONVENTIONS

A valid FILENAME consists of at most 10 characters starting with an alpha character.

The following characters may be used as wildcard specifiers.

? - matches any one character

* - matches any number of characters (including zero)

Wildcard specifiers allow one to perform operations on file families (e.g., CAT, ERASE, etc.).

If the first character of the filename is an ASCII 0 (zero), the filename will not appear in file listings when a CAT command is executed (see Section 5.3).

4.0 FILETYPES

The filetypes which may appear on a disk are:

1) BASIC - BASIC program
2) BYTES - memory image (saved using the CODE or SCREEN $ specification)
3) DATA - data file (any file created as a result of an OPEN command).
4) ARRAY - saved with the DATA specification

5.0 TS 2000 COMMAND SET

Notational Conventions:

```
     c = stream #    (0-15)
     m = device specification  ("m" for microdrive)
                               ("t" for text      )
                               ("b" for binary    )
                               ("n" for network   )
                               ("k" for keyboard  )
                               ("s" for screen    )
                               ("p" for printer   )
     n = drive number (1-8)
     r = record length (1-32767)
  [..] = optional item
```

Note: All of the commands below are case-insensitive.

5.1 OPEN #c ; "m" ; n ; "filename" [, "file type spec"]

OPEN #c; "n"; station_number
OPEN #c; "t"
OPEN # c; "b"

Opens a stream identified by c on microdrive n and creates a new file or opens an existing file as specified by "filename". For a description of "filename" see section 3.0.

"file type spec" = "I [r]" - INPUT only - in all cases 'r' optionally specifies the record length for random access files.

"R [r]" - INPUT/OUTPUT

"A [r]" - APPEND - automatically sets the file pointer to the end of the file (implicit in this specification is

If 'r' is not specified, then the file is opened for sequential access only. The default for filetype spec is "R".

If the device spec used is "k", "s" or "p" the delimiter following the stream number must be a comma. & nothing follows dev spec

5.2 CLOSE #c

Closes the file associated with the specified stream after flushing the appropriate buffers where necessary.

5.3 CAT n [; "filename"]

CAT #c ; n [; "filename"]

Causes a list of files on the specified microdrive n to be displayed on the screen in alphabetical order by filename. The second form causes the output to be directed to stream number c. The information displayed for each file is:

filename
filetype
date created
date last updated
size (in bytes)
number of free bytes on cartridge

The optional filename allows the listing of file families (see Section 3.0 above).

5.4 LIST #c

Causes the BASIC program currently contained in memory to be listed the stream number c.

5.5 FORMAT "m" ; n ; "volname"

FORMAT "n" ; number

FORMAT "b" ; baud rate

FORMAT "t" ; baud rate

"m" - Formats the cartridge on the specified microdrive and gives it the specified volume name.

"n" - Assigns the specified station number for addressing on the local network

"b" - ~~Opens a stream to~~ the RS 232 port with the specified baud rate.

"t" - ~~Opens a stream to~~ the RS 232 port with the specified baud rate.

The "t" channel (text) is used to transfer listings to a stream. The protocol for the various ASCII characters is as follows:

0 - 12	-	not sent
13	-	sends a carriage return and a line feed
14 - 31	-	not sent
32 - 127	-	sent as normal
128- 164	-	not sent ("?" sent instead)
165- 255	-	tokens are expanded as usual using ASCII codes 32 - 127.

Both INPUT and INKEY$ expect 7 bit characters from the "t" channel (8th bit is ignored.)

The "b" channel (binary) sends 8 bit codes.

Both INPUT and INKEY$ expect 8 bit characters from the "b" channel.

SAVE and LOAD work only with the "b" channel.

Line feed characters are not automatically inserted after carriage returns as they are for the "t" channel.

5.6 SAVE * (Note: '*' distinguishes from Tape SAVE)

SAVE * "m" ; n ; "filename"

Saves the current program and variables to the specified file on the specified drive.

```
SAVE  *  "m" ; n ;  "filename"  LINE
```

Saves the current program and variables to the specified file on the specified drive. When it is loaded, it automatically starts at the beginning of the program.

```
SAVE  *  "m" ; n ; "filename"  LINE  linenumber
```

Saves the current program and variables to the specified file on the specified drive. When it is loaded, it automatically starts at the specified linenumber.

```
SAVE  *  "m" ; n ; "filename"  DATA  arrayname  ()
```

Saves the specified numeric data array to to the specified file on the specified drive.

```
SAVE  *  "m" ; n ; "filename"  DATA  arrayname  $()
```

Saves the specified string data array to to the specified file on the specified drive.

```
SAVE  *  "m" ; n ; "filename"  CODE  start , len
```

Saves the byte array starting at address 'start' for a length of 'len' bytes to the specified file on the specified drive.

```
SAVE  *  "m" ; n ;  "filename"  SCREEN$
```

Saves the contents of the screen to the specified file on the specified drive. This is equivalent to:

```
SAVE  *  "m" ; n ; "filename"  CODE  16384 , 6912
```

Note that performing a SAVE command to channel "n" allows the program or data to be transferred to another station on the local network. The receiving station initiates the receive operation by performing the equivalent LOAD command (see Section 5.7). When this operation is used a filename is not specified and the receiving station number is specified as follows:

```
      SAVE * "n" ; station number
```

5.7 LOAD * (Note: '*' distinguishes from Tape LOAD)

```
LOAD  *  "m" ; n ; "filename"
```

Loads the current program and variables from the specified file on the specified drive. Any program already in memory is overwritten.

```
LOAD  *  "m" ; n ; "filename"  DATA  arrayname  ()
```

Loads the specified numeric data array from the specified file on the specified drive.

```
LOAD  *  "m" ; n ; "filename"  DATA  arrayname  $()
```

Loads the specified string data array from the specified file on the specified drive.

```
LOAD  *  "m" ; n ; "filename"  CODE  [ start [ , len ] ]
```

Loads the byte array into memory starting at address 'start' for a length of 'len' bytes from the specified file on the specified drive. If either 'start' or 'len' is not specified, the parameters with which the file was saved are used.

```
LOAD  *  "m" ; n ; "filename"  SCREEN$
```

Loads the contents to the screen from the specified file on the specified drive.

Note that performing a LOAD command from channel "n" allows the program or data to be transferred from another station on the local network. The sending station initiates the send operation by performing the equivalent SAVE command (see Section 5.6). When this operation is used a filename is not specified and the receiving station number is specified as follows:

```
LOAD * "n" ; station number
```

5.8 ERASE "m" ; n ; "filename"

Removes the specified file(s) from the specified drive.

```
            #C1                                      #C2
5.9   MOVE {"m" ; n ; "src filename"} TO {"m" ; n ; "dest filename"}
```

Causes the data at stream C1 or contained in the file identified by "src filename" to be output to stream C2 or copied to the file identified by "dest filename". Random access files may only be moved using the "src filename" and "dest filename" specifications.

5.10 VERIFY * (Note: '*' distinguishes from Tape VERIFY)

The following commands have the same semantics as the corresponding LOAD commands except that the data is not loaded into memory but compared with what is already there. An error message is displayed if they are not identical.

```
VERIFY  *  "m" ; n ; "filename"

VERIFY  *  "m" ; n ; "filename"  DATA  arrayname  ()

VERIFY  *  "m" ; n ; "filename"  DATA  arrayname  $()

VERIFY  *  "m" ; n ; "filename"  CODE [start [, len]]

VERIFY  *  "m" ; n ; "filename"  SCREEN$
```

Note that performing a VERIFY command from channel "n" allows the program or data currently in memory to be verified against that transferred from another station on the local network. The sending station initiates the send operation by performing the equivalent SAVE command (see Section 5.6). When this operation is used a filename is not specified and the receiving station number is specified as follows:

```
VERIFY * "n" ; station number
```

5.11 PRINT #c ; [AT rec # , byte #;] var list

Causes the variable list to be written to the file opened through stream c. The AT specification is used with random access files. It positions the file cursor at the specified byte number within the specified record. Note that individual items in the print stream must be separated by valid delimiters (comma or newline) and that strings must be written with surrounding quotes when their context requires it (see INPUT below).

5.12 INPUT #c ; [AT rec # , byte #] var list

Causes the variable list to be read from the file opened through stream c. The AT specification is described under PRINT (Section 5.11). Items are read from the file as if they were typed in from the keyboard (in the case of an INPUT statement) or from a DATA statement (using a READ statement).

5.13 MERGE * "m" ; n ; "filename" (Note: '*' distinguishes from Tape MERGE)

Loads the specified file from the specified drive, causing it to be merged with the BASIC program which is already in memory.

Note that performing a MERGE command from channel "n" allows the program or data to be transferred from another station on the local network. The sending station initiates the send operation by performing the equivalent SAVE command (see Section 5.6). When this operation is used a filename is not specified and the receiving station number is specified as follows:

```
MERGE * "n" ; station number
```

5.14 string var = INKEY$ #c

This causes the character at the current location in the file, at the RS 232 port or at the network to be assigned to the specified string variable.

6.0 ERROR CODES

SYNTAX ERROR

OPEN , SAVE , LOAD , VERIFY , MOVE

STREAM NOT OPEN

CLOSE , PRINT , INPUT , INKEY # , LIST , MOVE , CAT

STREAM ALREADY OPEN

OPEN

INVALID DEVICE SPECIFICATION

OPEN , CAT , FORMAT , SAVE , LOAD , ERASE , MOVE , VERIFY , MERGE

INVALID FILENAME

OPEN , SAVE , LOAD , ERASE , MOVE , VERIFY , MERGE , CAT

INVALID DRIVE NUMBER

FORMAT , CAT

FILE OPENED FOR SEQUENTIAL ACCESS

PRINT , INPUT

WRITE PROTECT ERROR

SAVE , ERASE , MOVE , PRINT

DEFECTIVE MEDIA

ALL DISK OPERATIONS

FILE NOT FOUND

LOAD , ERASE , MOVE , MERGE

FILE OPENED FOR READ ONLY

PRINT

CARTRIDGE FULL

SAVE , PRINT

END OF FILE

INPUT